# Resilience analysis of networked systems-of-systems based on structural and dynamic interdependencies

**Roberto Filippini**[a*], **Andres Silva**[b],
[a]Joint Research Centre of the European Commission, Ispra, Italy
[b]GIB Research Group, Facultad de Informática, Universidad Politécnica de Madrid, Spain

**Abstract:** Critical infrastructures support everyday activities in modern societies, facilitating the exchange of services and quantities of various nature. Their functioning is the result of the integration of diverse technologies, systems and organizations into a complex network of interconnections. Benefits from networking are accompanied by new threats and risks. In particular, because of the increased interdependency, disturbances and failures may propagate and render unstable the whole infrastructure network. This paper presents a methodology of resilience analysis of networked systems of systems. Resilience generalizes the concept of stability of a system around a state of equilibrium, with respect to a disturbance and its ability of preventing, resisting and recovery. The methodology provides a tool for the analysis of off-equilibrium conditions that may occur in a single system and propagate through the network of dependencies. The analysis is conducted in two stages. The first stage of the analysis is qualitative. It identifies the resilience scenarios, i.e. the sequence of events, triggered by an initial disturbance, which include failures and the system response. The second stage is quantitative. The most critical scenarios can be simulated, for the desired parameter settings, in order to check if they are successfully handled, i.e recovered to nominal conditions, or they end into the network failure. The proposed methodology aims at providing an effective support to resilience-informed design.

**Keywords:** System of systems, criticalities and vulnerabilities, interdependencies, resilience.

## 1. INTRODUCTION

Networks and modern infrastructures can be considered as special instances of systems of systems (SoS) [1]. A number of distinctive features characterize this class of systems. A list of these is provided in [2]: 1) operational and managerial independence, 2) geographical distribution, 3) emergent behavior and 4) evolutionary development. A fifth feature has to be added, especially in the case of networks and infrastructures; the interoperability, which establish interdependencies among the diverse components. These features makes SoS to escape from an easy classification into any of the existing engineering and theoretical frameworks, and require a solid interdisciplinary mindset. The crucial issue, to which many studies converge, is how to understand and govern risks and vulnerabilities in interdependent infrastructures [3, 4, 5, 6, 7, 8]. This is not a trivial issue and actually many of the existing approaches fail to return a solution. If one considers the risk assessment framework, for which the exact knowledge on system is prerequisite, the postulated emergent behavior and evolutionary development of the SoS would represent an hurdle for its application. Another hurdle is the specialization of the technology sectors that usually do not share a common analysis framework. Several contributions exist in this respect, see for example [9] and [10]. In the latter work, system interdependencies are modeled and simulated within the distributed environment High Level Architecture, which is based on the federation of diverse simulation platforms [11]. This modeling and simulation environment overcomes the technical diversity among systems, thus reproducing separately processes and quantities by means of the most suitable analysis tool. Nonetheless, with the exception of power grids and communication networks, this methodology cannot be easily scaled up or generalized. A significant rethinking out of the box is necessary, which concerns scope, objectives and the quantities to analyze [12, 13, 14].

In a framework for the modeling and analysis of SoS, interdependencies among components must be

within the scope. These are the preconditions for a SoS to operate, as well as they contribute to the propagation of disturbances and failures [15]. Systems have a partial view of (inter)dependencies, which is often limited to the components in the closest neighborhoods, those ones through which they establish a direct relationship. The more the knowledge on (inter)dependencies is incomplete, the more a system will become vulnerable from input disturbances. In its turn, and for similar reason, a failure in the same system may jeopardize the network. In resilience engineering this uncertainty on the system behavior is called system variability [3]. A consequence of system variability is that the boundary between system behavior and misbehavior is blurry. For instance, the response of a SoS may unpredictably evolve into diverse scenarios, and one of these could even be a dysfunctional failure that, instead of a fault, will be caused by a legitimate control action by one of its components. The problem, as it is formulated above, lays more in the domain of control rather than in the risk assessment. In this theoretical framework, the variability is the quantity to be controlled and the resilience is the system attribute that measures the success of this control action [3].

This paper presents a methodology for the resilience analysis of systems of systems. The scope of the methodology covers structural and dynamic (inter)dependencies that are established among systems via relationships of diverse nature, e.g. producer/consumer, provider/user, controller/controlled. The objectives are 1) to derive the dependency network 2) to model failure propagation through the dependency network, 3) to account for system variability within a resilience control paradigm (i.e. perturbation, leave from the nominal state and return to it after a transient) 4) to provide support to a resilience informed design for systems of systems and infrastructures. The methodology consists of two analysis tools, one for the structural analysis of criticalities and vulnerabilities, and one for the resilience analysis, both qualitative or quantitative. These tools make part of a modeling and analysis framework, which also includes an ad-hoc modeling language, presented in previous works [16, 17].

The paper consists of four sections. Section 2 presents the structural analysis of the dependency network. Section 3 presents the tools for the resilience analysis. Section 4 concludes the paper.

## 2. DEPENDENCY ANALYSIS

A system of systems is a collection of heterogeneous components that interact and cooperate by relationships of various nature, e.g. producer/consumer, provider/user, controller/controlled and so forth. In order to master complexity, the representation of the SoS requires some restriction in scope as well as adequate modeling abstraction. The idea is to limit the scope to the components' interface. Moreover, the relationships established at the interface are modeled as functional dependencies, dimension-free, so that the components involved in a dependency do not necessarily have to share the same domain of representation. The following definitions hold:

**Dependency network:** a dependency network of a SoS results from the transformation of relationships into functional dependencies. It is a directed graph G(N,A), with a set of nodes (N) and arcs (A).

**Dependency:** a functional dependency is a relationship between two nodes, either direct or indirect (i.e. mediated by other nodes).

An example of system of systems is shown in Figure 1. This example is an oversimplified representation of interoperability among three diverse infrastructures (gas, power grid and communication), and will serve as a proof of concepts along the paper. The model, depicted in the figure, is obtained with the Infrastructure Resilience-Oriented Modeling Language or IRML, see [16]. Five systems are represented: the gas network, the power plant, the transmission and the distribution systems, and the communication system. These systems exchange services and/or quantities among them. For example, the gas network supplies the power plant, which in its turn produces the electricity for the transmission

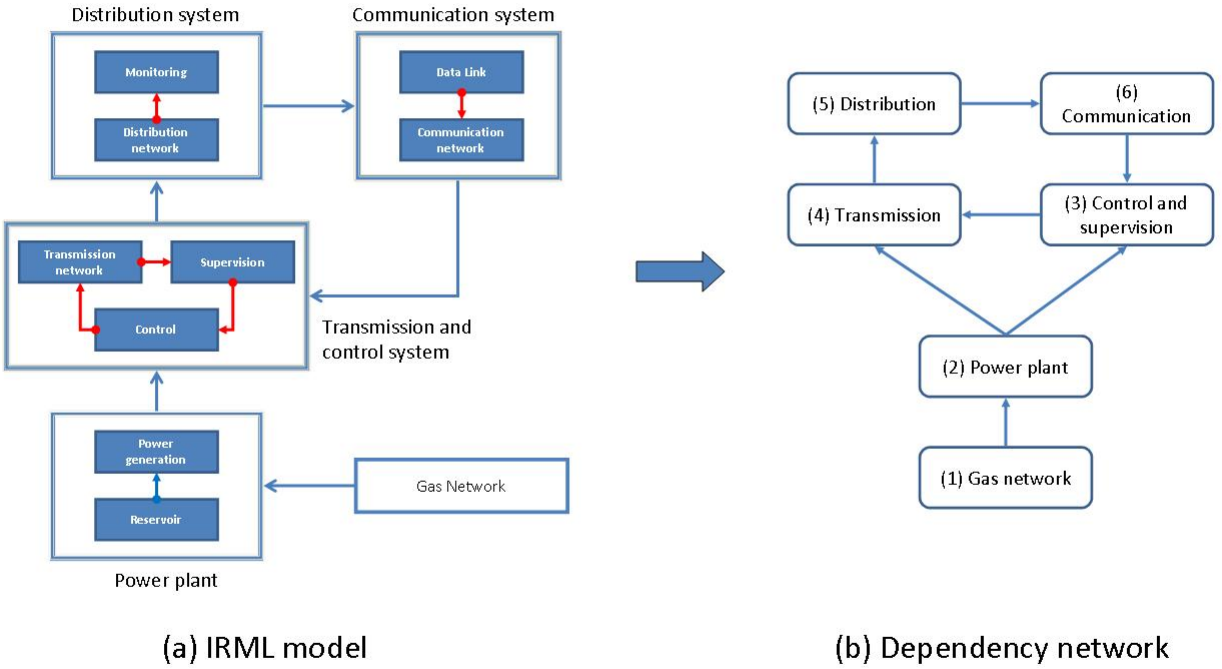


Figure 1: From the SoS to the dependency network.

system. The distribution system receives the electricity from the transmission and distributes it to the communication system. These are inter-system relationships, though there also intra-system relationships which are established among the different domains and resources that constitute a system. For example, control and supervision domains establish a control relationship over the transmission network. The reservoir in the power plant system is a resource for the power generation. Again, the monitoring in the distribution system supervises the load and transmits this information, via the communication data link, to the control system. In the real world, other dependencies are possible, but they are here ignored on purpose, for the sake of illustration.

The transformation of the IRML model into a dependency network is not strictly isomorphic, see also [17] for more details. Depending on the desired level of description, it is possible to give more emphasis to certain relationships and simplify others. In this example, we emphasize the dependencies on electricity consumption, the supply chain, and the control relationship established through the communication system. The result is shown in the dependency network of Figure 1(b). The six nodes represent the gas network (1), the power plant (2), the control and supervision (3), the transmission (4), the distribution (5) and the communication (6). Seven arcs account for the functional dependencies. Input arcs express the dependency of a node with respect to its ancestors, while output arcs express the dependencies of the descendant nodes with respect to the ancestor node. Forks and junctions are encountered in the dependency network, as well as loops. A fork means that more nodes will depend on the ancestor, for example nodes 3 (control) and 4 (transmission) depend on node 2 (power plant). A junction means that the node will depend on more ancestor nodes, which is the case of node 3 and 4. Loops will be examined in the following section.

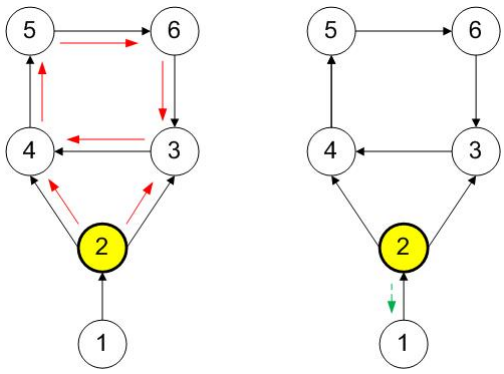


Figure 2: Criticality and vulnerability sets of node 2.

## 2.1. Analysis of the dependency network

The analysis of a dependency network returns the structural properties of specific node with respect to the other nodes, such as its criticality, vulnerability and the interdependency. But it may also provide aggregated metrics, such as the interaction and coupling coefficients of the network [6]. The following definitions hold:

**Criticality:** A node $k$ is critical with respect to nodes that depend on it. These nodes belong to the criticality set of node $k$, $C(k)$.

**Vulnerability:** A node $k$ is vulnerable with respect to the nodes on which it depends. These nodes belong to the vulnerability set of node $k$, $V(k)$.

Figure 2 shows the criticality and vulnerability sets of node 2 (power plant). This node has a larger criticality set [3, 4, 5, 6], though it is vulnerable (and depends on) only from node 1, the gas network. In the given model, this means that in case of failure of the power plant all nodes that transport, distribute, control and consume electricity will be affected, either directly or indirectly.

Two nodes that depend on each other are said to be interdependent. In literature, several definitions of interdependencies exist, and not only of functional nature, see for example [18]. The majority of interdependencies are indirect, that is they are mediated by other nodes. These interdependencies can be identified in the dependency network as it follows:

**Interdependency:** In the dependency network, two nodes are indirectly interdependent if they belong to a loop.

As a consequence of interdependency, the criticality and the vulnerability sets of a node are not disjoint. This implies that every node of the loop is both critical to and vulnerable from the other nodes in the loop. Figure 3 shows the criticality and vulnerability sets for node 3 (control), which belongs to the loop [3, 4, 5, 6]. The criticality and vulnerability sets of node 3 are not disjoint, as it was the case of node 2 in Figure 2. For instance, control and supervision (3) are both critical and vulnerable to communication, though the criticality is not by direct dependency, but mediated by the transmission and distribution systems.

Other metrics can be derived from the analysis of the dependency network. In particular, two metrics stand out: network *interaction* and *coupling* [6]. The network interaction is related to the network complexity and interdependencies. The presence of loops is the main factor that influences interactions among nodes. Accordingly, the higher the number of loops, the higher the interaction coefficient. Network coupling is proportional to the depth of the criticality set of a node, combined with its out-degree. The lower the depth, the lesser steps are necessary to recover the system when things go

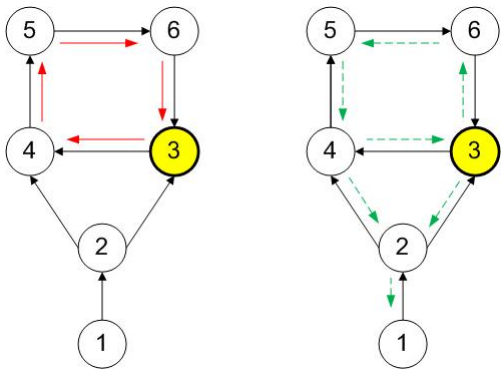


Figure 3: Criticality and vulnerability sets of node 3

wrong. Conversely, a larger node out-degree implies that the consequences of undesirable behavior from node k will influence more paths at a time.

These metrics are adapted from Perrow's theory for the analysis of complex systems [6]. According to these metrics, the worst situation would be that of a dependency network with many loops, i.e. strong interaction, and nodes with a low depth of the respective criticality sets and large out-degree. In contrast, a desirable situation would be described by a network with few loops and nodes with high depth and small out-degree. Given a dependency network, by the analysis of its graph, it will be possible to plot interaction vs. the coupling coefficient. Different graphs will lead to different classifications in this two dimension space, thus making it possible a comparison of their structural properties. For design purposes, given that a particular dependency network is analyzed with regard to Perrow's dimensions, and assuming that the network functionality is preserved, it is possible to say that the reduction of the interaction and coupling may contribute to improve the overall resilience.

## 3. RESILIENCE ANALYSIS

### 3.1. Assumptions and definitions

The criticality set of a node returns the maximum extension of a disturbance propagation in the dependency network, which is the worst scenario. More realistically, the network is expected to have measures in place to resist, stop propagation and recover to the initial state in a resilient fashion. The following assumptions govern the disturbance/failure propagation and the recovery dynamics throughout the network:

- A1: A disturbance affects one particular node and, from that node, it propagates to its closest descendants in the criticality set;
- A2: each node may fail (after a time to failure $T_F$), if it is affected by disturbance from at least one of the ancestor nodes;
- A3: each node may recover (after a time to recover $T_R$) if all ancestor nodes have recovered.

As a consequence of disturbance propagation, the nodes in the dependency graph change their state, from an assumed initial state. The recovery to this initial state, after a transient period, will attest that the network is resilient with respect to the given disturbance. The following definitions hold:

**Network state:** The state of the network is the set of the node's output states. The node output state is a binary variable that takes two values: *up* (1) if the node is functioning and *down* (0) if it has failed.

**Disturbance and failure:** a disturbance is the event that challenges a node to leave the *up* state to the *down* state. The failure is the transition of the node state from *up* to *down*.

The node change of state from 0 to 1 is due to performance degradation below a certain threshold, e.g. a minimum service level. This degradation process can be modeled internally to each node.

Every node is given two resilience measures: 1) resist to a disturbance and 2) recover from a failure. Buffering and/or spare components are examples of *resistance measures* in technical systems. These measures can withstand a disturbance for a given time interval before the node fails, i.e. the *time to failure* $T_F$. Recovery measures restore the node state to *up*, after a given time interval, i.e. the *time to recovery* $T_R$. Disturbance duration, time to failure and time to recovery are the parameters for the quantitative resilience analysis. These quantities are often available at service operators, and public utilities. For example, the power plant has got a reservoir to withstand the disruption in the gas supply chain, which can be estimated into several days, i.e. the $T_F$. It is important to remark that the failure of the power plant itself, caused by internal reasons, is not related to this $T_F$.

### 3.2. Resilience scenarios

A resilience scenario is the response of SoS to the failure propagation of a disturbance generated in one node, and to which it can be associated a trajectory in the space of states of the dependency network.

A resilience scenario may be either conditioned or unconditioned.

- **Unconditioned scenario**: a resilience scenario is unconditioned if, whatever the duration of the disturbance, when this stops, it is possible to recover back to the initial network state.
- **Conditioned scenario**: a resilience scenario is conditioned if the recovery back to the nominal network state is conditioned either by the duration of the disturbance or by structural properties.

The conditioned scenarios further specialize into deadlock scenarios and time-bounded scenarios.

- **Deadlock scenario**: a deadlock scenario occurs when all nodes in a loop have failed.
- **Time bounded scenario**: a time bounded scenario occurs when at least a deadline for recovery of a node exists, at the expiration of which that node(s) cannot be recovered anymore.

A deadlock cannot be removed once it is established in the loop, even though the initial disturbance stops. The removal of a deadlock would ask for additional resources (e.g. emergency management), which are out-of-the-loop, and not included in the dependency network. Deadlocks may be found in systems that produce a good and use a service which depends on the consumption of that good, directly or indirectly. In the given example, the communication depends on the electricity distribution, which depends on the transmission that is operated by the control system. This latter receives data from the communication, which trivially closes the loop.

A time bounded scenario occurs if the network is not recovered back to its initial state, by a given time interval. Time-bounded scenarios can be found in those systems that include parts that deteriorate or provide services that are non-interruptible. In the example, a rail freight transportation system that depends on the distribution and communication systems could be added. The fact that goods deteriorate if not delivered before a certain interval would make the scenario to be time-bounded.

In theory, a third type of resilience scenario exists. Failure and recovery events may run one after the other without ending up neither into a deadlock nor into the initial network state. This scenario is only possible with loops and a particular parameter settings. The resulting state trajectory will be unstable without leading to a network failure, i.e. a deadlock. Again, only an external intervention may prevent the network to fall into these scenarios.

### 3.3. Qualitative resilience analysis

The goal of the qualitative analysis is the generation of the resilience scenarios starting from an initial disturbance. This analysis is conceptually similar to model checking. Model parameters (disturbance duration, $T_F$ and $T_R$) do not need to be defined, being the analysis qualitative. The causes of the disturbance, and its nature, are also out of scope.

The generation of resilience scenarios starts from the node affected first by disturbance and continues by visiting the nodes in its criticality set. This process is similar to the building of an event sequence diagram in risk analysis, though it is conceptually different. The main difference is that events are here concurrent instead of being mutually exclusive, like *success* and *failure* are. Moreover, one branch for every concurring event has to be generated. In order to account for concurrency, each block of the diagram is labeled with two sets $F$ and $R$ that account respectively for the set of active failure events and the set of active recovery events. For an event to be in $F$ or $R$ means that the respective node is in the process of failing or recovering. The occurred failures and recovery events are associated to the transition arcs.

The concurrent event sequence diagram starts with the disturbance in the node k, which is the initiating event. As long as the disturbance persists, only failure events are active events (assumption A3). When the disturbance stops, recovery events are also enabled. The next event is selected from the list of the active concurrent events. In the case of a failure event: 1) the event is removed from $F$, 2) a new block is generated and 3) new failure events are added in $F$, which account for the descendant nodes affected by disturbance propagation. In a similar way, in the case of a recovery event, 1) this is removed from $R$, 2) a new block is generated and 3) new recovery events are added in $R$, which account for the descendant nodes that are enabled to recover. The sequence terminates if $R$ and $F$ are empty, and it is labeled as resilience scenario.

An example of resilience scenario generation is shown in Figure 4. The graphic convention states that the diagram develops downwards, if a failure occurs, and rightwards into a new column if a recovery occurs. The sequence is assumed to start with the failure of node 2 (power plant), i.e. the initiating event, which triggers disturbances at the same time in node 3 (control) and 4 (transmission). The failure propagates up to node 5 (distribution) before the disturbance from node 2 stops. After this event, node 3 recovers and enables the recovery of node 4 too. The sequence of events continues with the failure of node 6 (communication) and 3 (control) followed by the recovery of 5 (distribution), up to the point in which failure in 4 (transmission) and recovery in 6 (communication) are concurrent active events. From there on, the diagram takes two different directions. The early recovery of the communication network will result into a recovered scenario, while the early failure of the transmission will lead to a deadlock. This result is not necessarily the actual network behavior, but just a possible scenario. Still, it provides a punctual information on the sequence of events that may lead to the deadlock. This information should be dealt with care. The existence of diverse scenarios for a given disturbance is a consequence of the system variability. The quantitative analyses will make it possible to simulate the identified scenarios and check whether the network will end into a deadlock or it will recover, in a resilient fashion.

In the given example, only two scenarios are shown, for sake of illustration, but many more could have been generated. Many blocks in Figure 4 have more active events (in $F$ and $R$), from which diverse sequences could have departed. The outcome of the qualitative resilience is to identify all those sequences that lead to a deadlock, and this analysis should be done at least for the most critical nodes in the dependency network.

A significant reduction of scenarios is possible if heuristics are applied to cut branches and/or terminate a sequence. The number of nodes involved and the depth of propagation are two possible heuristics, related to the likelihood of an event sequence. For example, a long event sequence may be deemed to be unlikely if the events are independent. It is also possible to associate consequences to an event sequence, and this is another heuristic. If the estimated consequence are already non acceptable, from a certain event onwards, there will be no need to develop the scenario further.

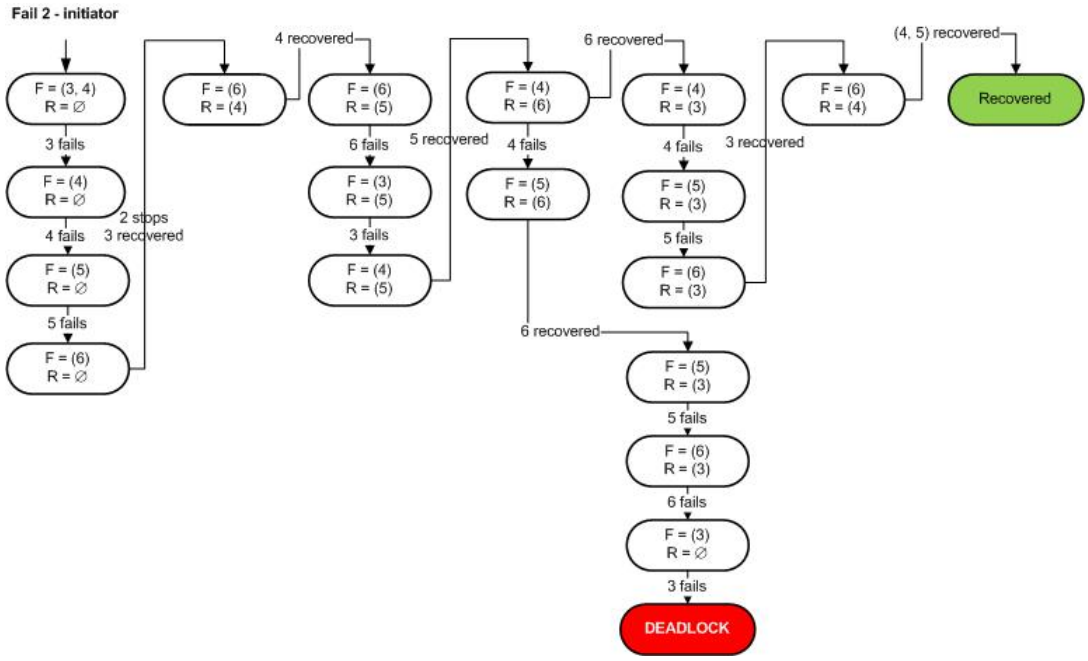


Figure 4: Two possible resilience scenarios: deadlock and recovered.

### 3.4. Quantitative resilience analysis

A resilience scenario can be associated to one or more trajectories in the network state space, depending on the parameters $T_F$ and $T_R$. A quantitative resilience analysis makes it possible to simulate this scenario, for the desired model parameter settings, and check whether it will end up into a recovered scenario, into a deadlock or into an unstable scenario. Given that for a dependency network the only acceptable state is the one for which all nodes are in the *up* state, the following definition of network resilience holds:

**Network resilience:** the network resilience is defined as the sum of all node's output states.

This definitions also applies to subsets of the network, like loops. Figure 5 shows the resilience of the loop [3, 4, 5, 6] versus time, i.e. $r = x_3+x_4+x_5+x_6$, for the deadlock and recovered scenarios identified in Figure 4. The analysis of the deadlock scenario is obtained with a default parameter setting for $T_F$ and $T_R$, equal to 1 for every node. The disturbance (a step function) occurs at t = 5, and last 2.5 time unit. Time units and the value of parameters do not refer to any real problem set up. The loop resilience shows an an irregular trend, during which the systems attempt the recovery but then they all fail, the loop resilience becomes zero and the loop enters the deadlock. The recovered scenario is obtained by halving the time to recovery of the node communication (6). Due to this modification, the recovery of communication is faster than the failure of the transmission, thus enabling earlier the recovery of control. In this scenario, the loop resilience moves back to its initial value ($r = 4$) after a transient of about 2.4 time units, and the deadlock is avoided. Details on the mathematics for the simulation are out of scope and they are omitted.

Qualitative and quantitative analysis are complementary tools, though the latter, by simulation of scenarios, makes it possible to obtain more insights for the resilience assessment. The scope is also larger. For instance, it is possible to model disturbances that affect the same node at periodic intervals, as well as multiple sources of disturbance that affect different nodes. The latter situation can be considered as an attack pattern, and it extends the scope of the analysis to security issues. In general, model parameters could be given probabilistic distributions. In this case, a Monte Carlo simulation will be the right tool.

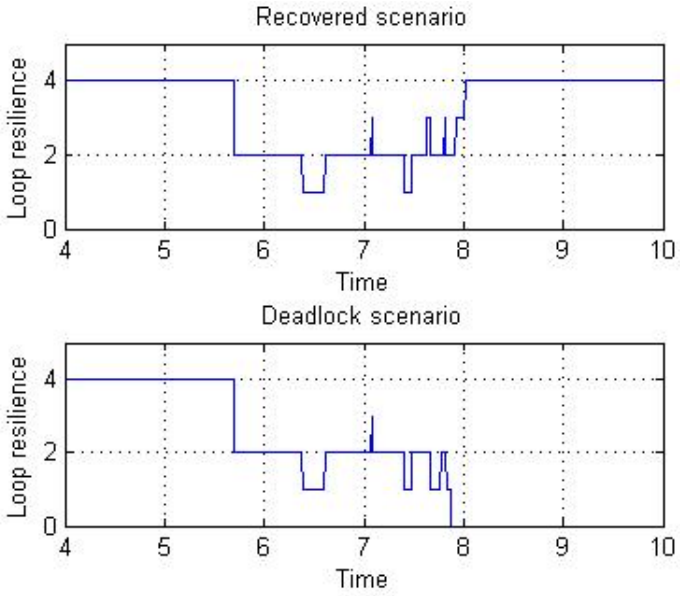


Figure 5: Simulation of resilience scenarios within the loop [3, 4, 5, 6].

## 4. CONCLUSIONS

This paper presented a methodology for the resilience analysis of networked systems of systems. The scope of the methodology covers structural and dynamic dependencies that establish among components, and are here represented by a dependency network. The analysis is both structural and dynamic. Structural analysis identifies criticalities and vulnerabilities in the dependency network. Dynamic analysis copes with the propagation of a disturbance, generated in one node, throughout the network. Resilience is analyzed in a qualitative way by identifying all sequences of events, caused by an initial disturbance, which end up into scenarios, either recoverable or non recoverable, i.e. deadlock. A quantitative analysis simulates these scenarios and evaluates the ability of the network to resist and recover in a resilient fashion. An example of interconnected, cooperating infrastructures was used as case study and proof of concepts along the paper.

The methodology introduces several original ideas that are here summarized. Relationships among SoS components are represented by a network of functional dependencies. The response to a disturbance is accommodated into a concurrent event sequence diagram. The definition of resilience takes inspiration from the concepts of stability and robustness in control system theory. Resilience analysis is conducted on a compact set of parameters that includes the disturbance duration, the times to failure and times to recovery of each model component. Finally, the analysis returns recommendations, which may be used by an operator or a decision maker, and integrated within a resilience informed design.

The analysis tools and the modeling language IRML are a stand-alone modeling framework, though they may also be applied together with risk assessment, should it be necessary to estimate the risk of a given scenario. Another application of the methodology, not presented in this paper, is in support to accident analysis, for discovering those vulnerabilities caused by (inter)dependencies. The modeling language IRML and the analysis tools will be implemented as software[1].

## REFERENCES

[1] A. Sousa-Poza, S. Kovacic, C. Keating, (2008) System of Systems Engineering: an Emerging Multidiscipline, *International Journal of System of Systems Engineering*, Vol. 1, pp. 1-17.

[1]Project supported by the Joint Research Centre, European Commission, 2011

[2] M. Jamshidi editor (2010), *System of Systems Engineering, Innovations for the 21st Century*, Wiley.

[3] E. Hollnagel, D. W. Woods, N. Leveson (Editors) (2006) *Resilience Engineering: Concepts And Precepts.* Ashgate.

[4] S. Jackson (2010), *Architecting Resilient Systems*, Wiley.

[5] N. Leveson (2004), A New Accident Model for Engineering Safer Systems, *Safety Science 42(4)*, pp. 237–270.

[6] C. Perrow (1999), *Normal Accidents: Living with High-Risk Technologies*, updated ed., Princeton University Press.

[7] J. P. G. Sterbenz, D. Hutchison, E. K. etinkaya, A. Jabbar, J. P. Rohrer, M. Schoeler et al., (2010) Resilience and Survivability in Communication Networks: Strategies, Principles, and Survey of Disciplines, *Computer Networks*, Vol 54, pp. 1245-1265.

[8] W. Kroeger, E. Zio, (2011) *Vulnerable Systems*, Springer.

[9] Bompard E., Napoli R. (2009), Xue F., Analysis of Structural Vulnerabilities in Power Transmission Grids, *International Journal of Critical Infrastructure Protection*, Elsevier, Vol. 2 pp. 5-12.

[10] C. Nan, I. Eusgeld (2011), Adopting HLA Standard for Interdependency Study, *Resilience Engineering and System Safety*, Vol. 96, Issue 1, January 2011, Pages 149-159.

[11] IEEE 1516 (2010), Standard for Modeling and Simulation High Level Architecture.

[12] Valerdi R, and others (2008), A Research Agenda for System of Systems Engineering, *Int. J. System of Systems Engineering*, Inderscience Publisher, Vol. 1, pp. 171-188.

[13] W. Kroeger, (2008) Critical Infrastructures at Risk: A Need for a New Conceptual Approach and Extended Analytical Tools, *Reliability Engineering and System Safety*, Vol. 93, pp. 1781 1787.

[14] M. J. Egan, (2007) Anticipating Future Vulnerability: Defining Characteristics of Increasingly Critical Infrastructure-like Systems, *Journal of Contingencies and Crisis Management*, Vol. 15, no. 1, pp. 4-17.

[15] S. Panzieri, R. Setola (2008). Failure Propagation in Critical Interdependent Infrastructures. *International Journal in Modeling identification and Control*, 3(1), pp. 6978.

[16] R. Filippini, A. Silva, (2011) Modeling Language for the Resilience Assessment of Networked Systems of Systems, Proceeding of the conference ESREL 2011, Troyes 18-22 Sept. 2011.

[17] A. Silva & R. Filippini (2011). Infrastructure (Resilience-oriented) Modeling Language: IRML. A Proposal for Modelling Infrastructures and their Connections. *JRC Scientific and Technical Reports*. JRC63302. JRC of the European Commission.

[18] S. M. Rinaldi, J. P. Peeremboom, T. K. Kelly (2001), Identifying, Understanding and Analyzing Critical Infrastructure Interdependencies. *IEEE Control Systems Magazine 21(6)*, pp. 11-25.